The Future of Edge Is About RISC-V and AI

Simon TC Wang
wangtc@andestech.com
Senior Technical Marketing Manager
Andes Technology
Feb. 22, 2023
The Diversity of AI Use-Cases

Vision
- Image classification
- Object detection
- Image segmentation
- Spoof detection
- Face unlock
- Eye tracking
- Avatar
- SLAM
- ...

Voice and Speech
- Audio front-end processing
- Keyword spotting
- Voice command
- Speech to text
- Natural language processing
- Text to speech
- ...

Any signal
- Sensor fusion with force, pressure, accelerometer, gyro, ampere meter, vibration, temperature, radar/lidar, sonar, ...
- Pattern recognition
- Predictive maintenance
- Healthcare
- ...

Subject to change without notice
copyright © 2021-2023 Andes Technology

Taking RISC-V® Mainstream
Andes Processors to Fit Your AI

Smart Camera
- Face trigger
- Object detection
- Positioning system
- Voice trigger
- Voice command
- Always-on

Smart IoT Devices
- Face unlock
- Bokeh
- Avatar
- Gesture recognition
- SLAM
- High resolution
- Voice trigger
- Object detection
- Pose detection
- Face unlock
- Face unlock

Smart Home
- Intelligent HMI
- Beamforming
- Speech to text
- Natural language
- Sensor fusion

Mobile
- ADAS, HDR
- > 10 cameras

Automotive
- ~30 MOPS
- ~100 MOPS
- > 1 GOPS
- > 1 TOPS
- > 10 TOPS
- >100 TOPS

Data center

Andes V-Series
(RV64GCPv)

Andes D/A-Series
(RV32/64GCP)

Andes N-Series
(RV32/64GC)

AnDLA I-Series

Taking RISC-V® Mainstream
Andes RISC-V Processors Family

**N-Series Baseline**
- RISC-V baseline 32/64-bit
- + AndeStar™ V5 ext.
  - (CoDense™, Performance)
- 2 to 13 stage pipelines
- Single issue, superscalar
- In-order, out-of-order
- SMP, cache, local memory, ECC, ...

**D/A-Series DSP/SIMD**
- RISC-V baseline 32/64-bit
- + AndeStar™ V5 ext.
- + RISC-V Packed-SIMD ext.
  - (RVP draft)
- MMU (A-Series)
- SIMD width: 32, 64
- Data types: INT8/16/32

**V-Series Vector**
- RISC-V baseline 64-bit
- + AndeStar™ V5 ext.
- + RISC-V Vector ext. (RVV)
- VLEN/SIMD width: 128-1024
- LMUL (Length Multiplier): 1-8
- Data types: INT4/8/16/32/64, BF16, FP16/32/64

**AnDLA**
- Accelerated engine for GEMM, MAC accumulator, element-wise, pooling, etc
- Dedicated DMA and local shared SRAM

---

- Speedup and code size reduction with AndeStar™ V5¹
  - Performance (CoreMark/Dhrystone)
  - Code size (CSiBE)
  - +13% -12%

- Speedup with RVP draft²
  - CIFAR-10 Image classification (RV64P)
  - ML-KWS Keyword spotting (RV32P)
  - 14.8x 8.9x

- Speedup with RVV³
  - Conv Q7 33x33x51
  - GEMM FP32 32x32
  - 60.7x 49.8x

- Speedup with AnDLA⁴
  - VGG19 INT8
  - 263x

---

1: 25-Series, FPGA, Andes toolchain over open-source GCC v7.4
2: 25-Series, FPGA
3: NX27V, FPGA, (VLEN,SIMD)=(512,512), L1D 512 KB
4: I350, 64 MAC, FPGA
Andes RISC-V DSP/SIMD Processors

- **AndesCore™ D25F**
  - RV32GC/P*/B + AndeStar™ V5 ext.
  - 5-stage in-order single-issue

- **AndesCore™ D45**
  - RV32GC/P* + AndeStar™ V5 ext.
  - 8-stage in-order dual-issue
  - MemBoost
    - Instruction and data cache pre-fetch
    - Non-blocking loads/stores
    - Data cache write-around
    - Optional separated BIU I/D buses

- **AndesCore™ common technologies**
  - Hardware misaligned access, CoDense™, PowerBrake, QuickNap™, ...
  - Andes Custom Extension (ACE)

### Features

<table>
<thead>
<tr>
<th>Features</th>
<th>D25F</th>
<th>D45 (over D25F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoreMark/MHz(^1)</td>
<td>3.57</td>
<td>5.67 (+59%)</td>
</tr>
<tr>
<td>DMIPS/MHz (no-inline)(^1)</td>
<td>1.98</td>
<td>2.86 (+44%)</td>
</tr>
<tr>
<td>Gate count(^2)</td>
<td>100%</td>
<td>140% (+40%)</td>
</tr>
</tbody>
</table>

---

\(^1\): For AndeSight 5.0 toolchain, COPTS= -O3 -flto -fno-inline; CoreMark: AndeSight 5.0 toolchain, OPTFLGS= -O3 -funroll-all-loops -finline-limit=600 -fno-dominator-opts -fno-if-conversion2 -fselective-scheduling -fno-code-hoisting.

\(^2\): TSMC 7nm Fin FET High Speed L1 Cache Memory Compiler, Frequency condition: worst: SSGNP/0.675V/-40°C; Dynamic power condition: TT/0.75V/85°C.

---

\*: RVP (draft)
# Use Case: CIFAR-10 Image Classification

<table>
<thead>
<tr>
<th>CIFAR-10</th>
<th>Operators</th>
<th>Andes NN Library APIs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer 1</td>
<td>Convolution</td>
<td>riscv_nn_conv_HWC_s8_s8_s8_RGB_sft_bias_fast</td>
</tr>
<tr>
<td></td>
<td>Activation (ReLU)</td>
<td>riscv_nn_relu_s8</td>
</tr>
<tr>
<td></td>
<td>Pooling (maxpool)</td>
<td>riscv_nn_maxpool_HWC_s8</td>
</tr>
<tr>
<td>Layer 2</td>
<td>Convolution</td>
<td>riscv_nn_conv_HWC_s8_s8_s8_s8_sft_bias_fast</td>
</tr>
<tr>
<td></td>
<td>Activation (ReLU)</td>
<td>riscv_nn_relu_s8</td>
</tr>
<tr>
<td></td>
<td>Pooling (maxpool)</td>
<td>riscv_nn_maxpool_HWC_s8</td>
</tr>
<tr>
<td>Layer 3</td>
<td>Convolution</td>
<td>riscv_nn_conv_HWC_s8_s8_s8_s8_sft_bias_fast</td>
</tr>
<tr>
<td></td>
<td>Activation (ReLU)</td>
<td>riscv_nn_relu_s8</td>
</tr>
<tr>
<td></td>
<td>Pooling (maxpool)</td>
<td>riscv_nn_maxpool_HWC_s8</td>
</tr>
<tr>
<td>Layer 4</td>
<td>Fully-connected</td>
<td>riscv_nn_fc_s8_s8_s8_s8_sft_bias_fast</td>
</tr>
<tr>
<td>Layer 5</td>
<td>Softmax</td>
<td>riscv_nn_softmax_s8_fast</td>
</tr>
</tbody>
</table>

## Speedup of CIFAR-10

The higher the better

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>as the base</th>
</tr>
</thead>
<tbody>
<tr>
<td>M4</td>
<td>1.0x</td>
<td></td>
</tr>
<tr>
<td>M7</td>
<td>1.2x</td>
<td></td>
</tr>
<tr>
<td>Andes RV32P D25F</td>
<td>2.9x</td>
<td></td>
</tr>
<tr>
<td>Andes RV32P D45</td>
<td>3.7x</td>
<td></td>
</tr>
</tbody>
</table>

The CAT is classified as a cat.
Andes RISC-V Vector Processors

- **AndesCore™ NX27V**
  - RV64GC/P1/V2 + AndeStar™ V5 ext.
  - 5-stage in-order single-issue scalar unit
  - Vector Processing Unit (VPU)
    - INT4/8/16/32/64, FP16/32/64, BF16
    - VLEN & SIMD: 128/256/512, 1:1 or 2:1

- **AndesCore™ AX45MPV**
  - RV64GC/B/P1/V + AndeStar™ V5 ext.
  - 8-stage in-order dual-issue scalar unit
  - Vector Processing Unit (VPU)
    - Dual-issue vector pipeline
    - INT4/8/16/32/64, FP16/32/64, BF16
    - VLEN & SIMD: 128/256/512/1024, 1:1 or 2:1

- **AndesCore™ common technologies**
  - MemBoost memory subsystem
  - ACE (Andes Custom Extension)
  - ACE Streaming Port (ASP), ACE-RVV³

---

1: RVP draft
2: RVV v1.0 except segment load/store
3: 45V only
## Use Case: MobileNet-v1 Person Detection

The lower the better

<table>
<thead>
<tr>
<th>MobileNet-v1</th>
<th>Data Type</th>
<th>ISA</th>
<th>VLEN (bit)</th>
<th>SIMD (bit)</th>
<th>Normalized latency (ms @1GHz, 1 core)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA9 (Xilinx PYNQ)¹</td>
<td>FP32</td>
<td>NEON</td>
<td>-</td>
<td>128</td>
<td>703.4</td>
</tr>
<tr>
<td>CA53 (Raspberry Pi-3B)¹</td>
<td>FP32</td>
<td>NEON</td>
<td>-</td>
<td>128</td>
<td>438.5</td>
</tr>
<tr>
<td>CA72 (Firefly RK3399)¹</td>
<td>FP32</td>
<td>NEON</td>
<td>-</td>
<td>128</td>
<td>210.3</td>
</tr>
<tr>
<td>CA73 (Kirin 970)¹</td>
<td>FP32</td>
<td>NEON</td>
<td>-</td>
<td>128</td>
<td>292.4</td>
</tr>
<tr>
<td>Andes NX27V²</td>
<td>FP16</td>
<td>RVV</td>
<td>128</td>
<td>128</td>
<td>127.6</td>
</tr>
<tr>
<td>Andes NX27V²</td>
<td>FP16</td>
<td>RVV</td>
<td>256</td>
<td>128</td>
<td>100.5</td>
</tr>
<tr>
<td>Andes NX27V²</td>
<td>FP16</td>
<td>RVV</td>
<td>256</td>
<td>256</td>
<td>69.9</td>
</tr>
<tr>
<td>Andes NX27V²</td>
<td>FP16</td>
<td>RVV</td>
<td>512</td>
<td>256</td>
<td>56.9</td>
</tr>
<tr>
<td>Andes NX27V²</td>
<td>FP16</td>
<td>RVV</td>
<td>512</td>
<td>512</td>
<td>42.2</td>
</tr>
</tbody>
</table>

1: TVM, https://github.com/apache/tvm/wiki/Benchmark#arm-cpu
2: Andes libnn, PyTorchCV imagenet-1k “MobileNet x1.0” (https://pypi.org/project/pytorchcv/), based on FPGA and scaling to 1.0 GHz. Real SoC performance will depend on memory subsystem
**Andes Deep Learning Accelerator (AnDLA)**

- A standalone deep learning accelerator for edge inference
  - Scalable and multicore accelerator
  - Cooperate with AndesCore™ 25/27/45/60-Series

- Supported NN models
  - **Video and image:** VGG, Mobilenet, ResNet, YOLO, SSD, Inception, GooLeNet, DenseNet, ...
  - **Audio and voice:** RNN, LSTM, GRU, ...

- Accelerated NN operators
  - Convolution, fully-connected, activation, pooling, depth-wise, element-wise, ...
  - Operator fusion

- Target performance
  - Configurable MACs: 32 to 2048 (INT8)
  - Performance: 64 GOPS to 4 TOPS (INT8 @1GHz)
  - Leading power efficiency >5 TOPS/W (@28nm)

- Integrated DMA and local shared memory

---

1: the preliminary target on current roadmap; VGG and fixed 64 MACs only in 2022
Efficient Engine for NN Computation

Convolution kernel
(from CIFAR-10 layer 2)

<table>
<thead>
<tr>
<th>Baseline</th>
<th>RV64P</th>
<th>AnDLA (+baseline)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1x</td>
<td>19x</td>
<td>360x</td>
</tr>
</tbody>
</table>

Gate counts

<table>
<thead>
<tr>
<th>Baseline</th>
<th>RV64P</th>
<th>AnDLA (+baseline)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1x</td>
<td>2.2x</td>
<td>2.3x</td>
</tr>
</tbody>
</table>

Note
- preliminary data based on 25-Series
- AnDLA with 64 MACs configuration
- Gate counts @28nm HPC+; AnDLA logic only and without shared memory; CPU logic only and without L1 cache
## Efficient Engine for NN Computation

### ResNet-50 Performance

<table>
<thead>
<tr>
<th></th>
<th>NX27V (RVV 512/256 + VDOT)</th>
<th>AnDLA I350 (128 MAC)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Gate Counts</strong></td>
<td>1.00x</td>
<td>0.27x</td>
</tr>
<tr>
<td><strong>Performance Efficiency</strong></td>
<td>57x</td>
<td>1.00x</td>
</tr>
</tbody>
</table>

### Gate Counts

1. AnDLA performance doesn't include softmax which is around 0.009% of total cycles and it could be ignored; AnDLA results is estimated from tools plus the estimated error from Tiny YOLO experience between FPGA and tools.
2. TSMC N7 (SVT, LVT, ULVT) 240H library; AXI BUS with I/O constrain, synthesis with 30% of clock period margin; frequency condition: 0.675v/-40oc, SS; area condition: 65% utilization with scan; dynamic power condition: 0.75v/25oc, TT. Pre-layout simulation; RVV, logic only, 128-entry BTB, 16-entry PMP/PMA and 32KB I $; AnDLA, logic only, 128 MAC.
Andes Extensible RISC-V AI Subsystem

Processing Element (PE)
- Vector Extensible Processor
  - Scalar
  - Memory Subsystem
- Vector Unit
- ACE-RVV
- ACE
- ACE Streaming Port

AnDLA
- Compute
- DMA
- SRAM

Extended Type 1. Computing power
Extended Type 2. Data exchange
Extended Type 3. Control signal

1: the preliminary target on current roadmap; ACE, ACE-RVV, ASP and ACE-AnDLA are separated add-on packages

Taking RISC-V® Mainstream
AndeSight™ IDE
&
Andes NN SDK
Andes Tools and Software Stacks for AI

- **AndesSight™ IDE**
  - GCC/LLVM Toolchains
  - Build, debug, deploy, profile
  - Analysis and tuning
  - RTOS & Linux
  - Device drivers
  - Sample codes
  - Simulator
  - Documentation

- **Signal and data processing**
  - pre-/post-processing for Audio/voice, video/image, ...

- **Andes AI Offline Optimizer**

- **AI frameworks & model formats**
  - TensorFlow
  - PyTorch
  - ONNX

- **NN inference engines**
  - TensorFlow Lite
  - TensorFlow Lite
  - TVM

- **Andes LLVM compiler**

- **AndesCore™ N/D/A/V-Series**
  - AnDLA I-Series

- **AI applications**

- **Andes Tools and Software Stacks for AI**
  - AndeSoft™ NN Library
  - AndeSoft™ Vector Library
  - AndeSoft™ DSP Library
  - AndeSoft™ NN Library

Subject to change without notice

Copyright © 2021-2023 Andes Technology
Andes NN SDK

- Andes AI Offline Optimizer
  - Pruning
  - Operator fusion
  - Compensation
  - Calibration
  - Quantization
  - Memory/Operator allocation and optimization

- Front-end Converter
- Network Optimizer
- Deployment Optimizer
- Backend Converter

- Generated C code template
- TFL/TFLM

- Andes NN library
  - AnDLA driver

- RV Baseline
- RVP (draft)
- RVV
- AnDLA
Inference Flow with TensorFlow Lite for MCU

**Host (offline)**
- TensorFlow framework
- NN model .tflite file
- Offline optimizer
  - Pruning
  - Quantization
  - ...

**Device (runtime)**
- TensorFlow Lite for MCU runtime
- TenforFlow Lite for MCU reference kernels
- AndeSoft™ NN Library optimized kernels
- AnDLa driver
- AndesCore™ Baseline
- AndesCore™ RVP/RVV
- AnDLa

### TFLM kernel

<table>
<thead>
<tr>
<th>TFLM kernel</th>
<th>TFLM reference implementation</th>
<th>Libnn for RVP/RVV</th>
<th>AnDLa driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONV_2D</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>FULLY_CONNECTED</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SOFTMAX</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>EXP</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- TFLM reference kernels (fallback; pure-C implementation) are always available.
- Switch to hardware architecture with higher performance if possible (at compiling time).
Andes Readiness for TFL Models

- TensorFlow Hub: runnable percentage > 93% in models of Image, Text, Video, Audio domains\(^1\)
  - Using TF Flex operators\(^2\) and proprietary custom operators are the most of reasons causing failure

<table>
<thead>
<tr>
<th>Domain</th>
<th>Sub-Domain</th>
<th>Valid Models (excluded TPU)</th>
<th>Successfully Executed</th>
<th>Executable Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td>Classification</td>
<td>107</td>
<td>107</td>
<td>100.0%</td>
</tr>
<tr>
<td></td>
<td>Segmentation</td>
<td>18</td>
<td>18</td>
<td>100.0%</td>
</tr>
<tr>
<td></td>
<td>Object Detection</td>
<td>33</td>
<td>30</td>
<td>90.9%</td>
</tr>
<tr>
<td></td>
<td>Pose Detection</td>
<td>8</td>
<td>8</td>
<td>100.0%</td>
</tr>
<tr>
<td></td>
<td>Super Resolution</td>
<td>5</td>
<td>5</td>
<td>100.0%</td>
</tr>
<tr>
<td></td>
<td>Others</td>
<td>30</td>
<td>24</td>
<td>80.0%</td>
</tr>
<tr>
<td>Text</td>
<td></td>
<td>8</td>
<td>5</td>
<td>62.5%</td>
</tr>
<tr>
<td>Video</td>
<td></td>
<td>6</td>
<td>6</td>
<td>100.0%</td>
</tr>
<tr>
<td>Audio</td>
<td></td>
<td>16</td>
<td>12</td>
<td>75.0%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>231</strong></td>
<td><strong>215</strong></td>
<td><strong>93.1%</strong></td>
</tr>
</tbody>
</table>

Note
1: Models from TensorFlow Hub, Nov. 2022, [https://tfhub.dev/](https://tfhub.dev/)
2: A TF experiment features: need full TF runtime and not suitable for edge devices
AndeSoft™ NN Library

- Optimized neural network functions for RVP (draft) and RVV processors
- Boost NN performance by using SIMD and Vector instructions
- >170 functions in 8 categories
- Superset and compatible with CMSIS-NN APIs

Average speedup of NN library

<table>
<thead>
<tr>
<th>Model</th>
<th>Activation</th>
<th>Basic math</th>
<th>Concatenation</th>
<th>Convolution</th>
<th>Fully-connected</th>
<th>Pooling</th>
<th>Softmax</th>
<th>Utils</th>
</tr>
</thead>
<tbody>
<tr>
<td>N25F</td>
<td>1.0x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N45</td>
<td>1.6x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D25F RV32P</td>
<td>6.7x</td>
<td>11.3x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AX25 RV64P</td>
<td>8.4x</td>
<td>13.9x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D45 RV32P</td>
<td>29.0x</td>
<td>34.7x</td>
<td>34.7x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AX45 RV64P</td>
<td>48.7x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NX27V RVV (256/256/256)*</td>
<td>11.3x</td>
<td>29.0x</td>
<td>29.0x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NX27V RVV (512/256/256)*</td>
<td>13.9x</td>
<td>34.7x</td>
<td>34.7x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NX27V RVV (512/512/512)*</td>
<td>13.9x</td>
<td>48.7x</td>
<td>48.7x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1: separated product packages needed additional licenses
2: compare with full APIs; data based on FPGA
*: (VLEN/SIMD/BIU)
Commit to Compute Library Evolution

- **AndeSoft™ DSP Library**
  - Optimized for **RVP DSP/SIMD** processors
  - >320 functions in 10 categories: basic, complex, controller, distance, filtering, matrix, sort, statistic, svm, transform, utils
  - Superset and compatible with CMSIS-DSP API

- **AndeSoft™ Vector Library**
  - Optimized for **RVV Vector** processors
  - > 200 functions in 5 categories: basic, filtering, image, matrix, and transform
  - Superset and compatible with NE10 library APIs

The total function number of Andes NN / DSP/ Vector libraries

<table>
<thead>
<tr>
<th>Before</th>
<th>After</th>
<th>Number of newly added functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>~560 (ASTv510 @2022/2/18)</td>
<td>~690 (ASTv520 @2023/1/13)</td>
<td>+130</td>
</tr>
<tr>
<td>~690 (ASTv520 @2023/1/13)</td>
<td>~870 (ASTv530 @2023/Q4)</td>
<td>+180</td>
</tr>
</tbody>
</table>
Summary

- Andes RISC-V DSP/SIMD and Vector processors provide highly efficient computing power for diversified AI applications and segments.

- Andes is the leading pioneer to deliver the RISC-V extensible architecture with Andes Custom Extension (ACE), ACE-RVV, ACE Streaming port (ASP).

- AnDLA plays a key role to bring the power-efficiency computing to the "Andes Extensible RISC-V AI Subsystem".

- AndeSight™ IDE and Andes NN SDK brings the ultimate runtime performance and development efficiency to developers.